Docket No. AUS920030696US1 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Johns et al.**	§	Group Art Unit: **2181**
	§	
Serial No. **10/809,553**	§	Examiner: **Lee, Chun Kuan**
	§	
Filed: **March 25, 2004**	§	Customer No.: **50170**
	§	
For: **Method to Provide Cache Management Commands for a DMA Controller**	§	
	§	
	§	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

APPELLANTS' BRIEF (37 C.F.R. § 41.37)

This Appeal Brief is in furtherance of the Notice of Appeal filed July 10, 2007 (37 C.F.R. § 41.31).

The fees required under § 41.20(b)(2), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying Transmittal of Appeal Brief.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals and interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

The status of the claims involved in this proceeding is as follows:

1. Claims canceled: NONE
2. Claims withdrawing from consideration but not canceled: NONE
3. Claims pending: 1-21
4. Claims allowed: NONE
5. Claims rejected: 1-21

The claims on appeal are: claims 1-21

STATUS OF AMENDMENTS

No amendments to the application were filed subsequent to mailing of the Final Office Action.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent Claim 1:

The present invention provides a system to provide software program control of cache management. See specification, page 2, lines 14-20, for example. The system comprises a processor (110, for example) configured to generate direct memory access (DMA) commands for the management of a cache on the execution of a software program on the processor. See specification, page 4, line 14, to page 5, line 23; page 6, lines 10-21; page 10, lines 10-17; 302 in FIG. 3; 402 in FIG. 4, for example. The system further comprises a DMA controller (150, 200, for example) coupled to the processor that is configured to execute the DMA commands for the management of a cache. See specification, page 5, line 24, to page 6, line 21; 304-312 in FIG. 3; 404-418 in FIG. 4, for example.

Independent Claim 9:

The present invention provides a method for cache management in a system comprising a DMA controller (150, 200, for example) and a processor (110, for example). See specification, page 2, lines 14-20, for example. The method comprises running software on the processor to generate DMA commands for management of a cache. See specification, page 4, line 14, to page 5, line 23; page 6, lines 10-21; page 10, lines 10-17; 302 in FIG. 3; 402 in FIG. 4, for example. The method further comprises issuing the DMA commands to the DMA controller and executing the DMA commands. See specification, page 5, line 24, to page 6, line 21; 304-312 in FIG. 3; 404-418 in FIG. 4, for example.

Independent Claim 15:

The present invention provides a computer program product for cache management in a system comprising a DMA controller (150, 200, for example) and a processor (110, for example). See specification, page 2, lines 14-20, for example. The computer program product comprises a medium with a computer program embodied thereon. See specification, page 3, lines 16-23, for example. The computer program comprises computer code for running software on the processor to generate DMA commands for management of a cache. See specification, page 4, line 14, to page 5, line 23; page 6, lines 10-21; page 10, lines 10-17; 302 in FIG. 3; 402 in

FIG. 4, for example. The computer program further comprises computer code for issuing the DMA commands to the DMA controller and computer code for executing the DMA commands. See specification, page 5, line 24, to page 6, line 21; 304-312 in FIG. 3; 404-418 in FIG. 4, for example.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection on appeal are as follows:

I. Claims 1, 3, 9, 11, and 15-17 are rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by *Futral et al.* (U.S. Patent Publication No. 2005/0033874).

II. Claims 2, 7, 8, 10, 13, 14, 20, and 21 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Futral et al.* in view of *Ollivier et al.* (U.S. Patent No. 6,738,881).

III. Claims 4, 5, 12, 18, and 19 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Futral et al.* in view of *Liao et al.* (U.S. Patent No. 6,681,296).

IV. Claim 6 is rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Futral et al.* in view of *Ohba* (U.S. Patent No. 6,427,201).

ARGUMENT

I. 35 U.S.C. § 102, Alleged Anticipation of Claims 1, 3, 9, 11, and 15-17

The Office rejects claims 1, 3, 9, 11, and 15-17 under 35 U.S.C. § 102(e) as allegedly being anticipated by *Futral et al.* (U.S. Publication No. 2005/0033874). This rejection is respectfully traversed.

Futral teaches direct memory access using memory descriptor list. An operating system may allocate pages to a buffer and may build a memory descriptor list that references the pages allocated to the buffer. A direct memory access (DMA) controller may process the memory descriptor list and transfer data between a buffer defined by the memory descriptor list and another location per the memory descriptor list. See *Futral*, Abstract.

In contradistinction, the present invention, as recited in claims 1, 9, and 15, for example, provides a processor that is configured to generate DMA commands for the management of a cache on the execution of a software program on the processor. The DMA controller is coupled

to the processor and is configured to execute the DMA commands for the management of a cache.

With respect to claims 1, 9, and 15, the Final Office Action alleges that *Futral* teaches a processor that generates DMA commands for the management of a cache, and that the DMA controller is configured to execute the DMA commands for the management of a cache at paragraphs [0001], [0011], and [0015]. The cited portions of *Futral* are as follows:

> [0001] In a computing device, a processor may offload a data transfer to a direct memory access (DMA) engine or controller. In response to a data transfer request, the processor may execute a device driver. The device driver may cause the processor to generate one or more DMA descriptors defining the data transfer. The DMA controller may then process the DMA descriptors and transfer data per the DMA descriptors.
>
> [0011] Embodiments of the invention may be implemented in hardware, firmware, software, or any combination thereof. Embodiments of the invention may also be implemented as instructions stored on a machine-readable medium, which may be read and executed by one or more processors. A machine-readable medium may include any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computing device). For example, a machine-readable medium may include read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other forms of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.), and others. Further, firmware, software, routines, and/or instructions may be described herein as performing certain actions. However, it should be appreciated that such descriptions are merely for convenience and that such actions in fact result from computing devices, processors, controllers, or other devices executing the firmware, software, routines, instructions, etc.
>
> [0015] The chipset 112 may also comprise a direct memory access (DMA) engine or controller 122. The DMA controller 122 may process a DMA command structure 124 and perform data transfers that involve buffers 126 and/or I/O devices 120 per the DMA command structure 124. In one embodiment, the buffers 126 may be virtually contiguous but physically non-contiguous. It should be appreciated that virtually contiguous buffers 126 may permit software and firmware modules such as, for example, application 108 to easily manipulate the buffer 126 via a range of sequential virtual addresses. Further, not requiring the buffer 126 to be physically contiguous may enable the operating system 104 to

more effectively manage the memory 116. However, the DMA controller 122 may be limited to accessing memory 116 via physical addresses. Accordingly, the DMA command structure 124 directs the DMA controller 122 to the potentially physically non-contiguous buffer fragments 128 that comprise the buffers 126.

The cited portions of *Futral* fail to teach or suggest DMA commands for the management of a cache. In fact, the word "cache" does not appear anywhere in the *Futral* reference.

The Final Office Action alleges that *Futral* teaches a cache at reference number 116 in FIG. 1. The cited portion is as follows:

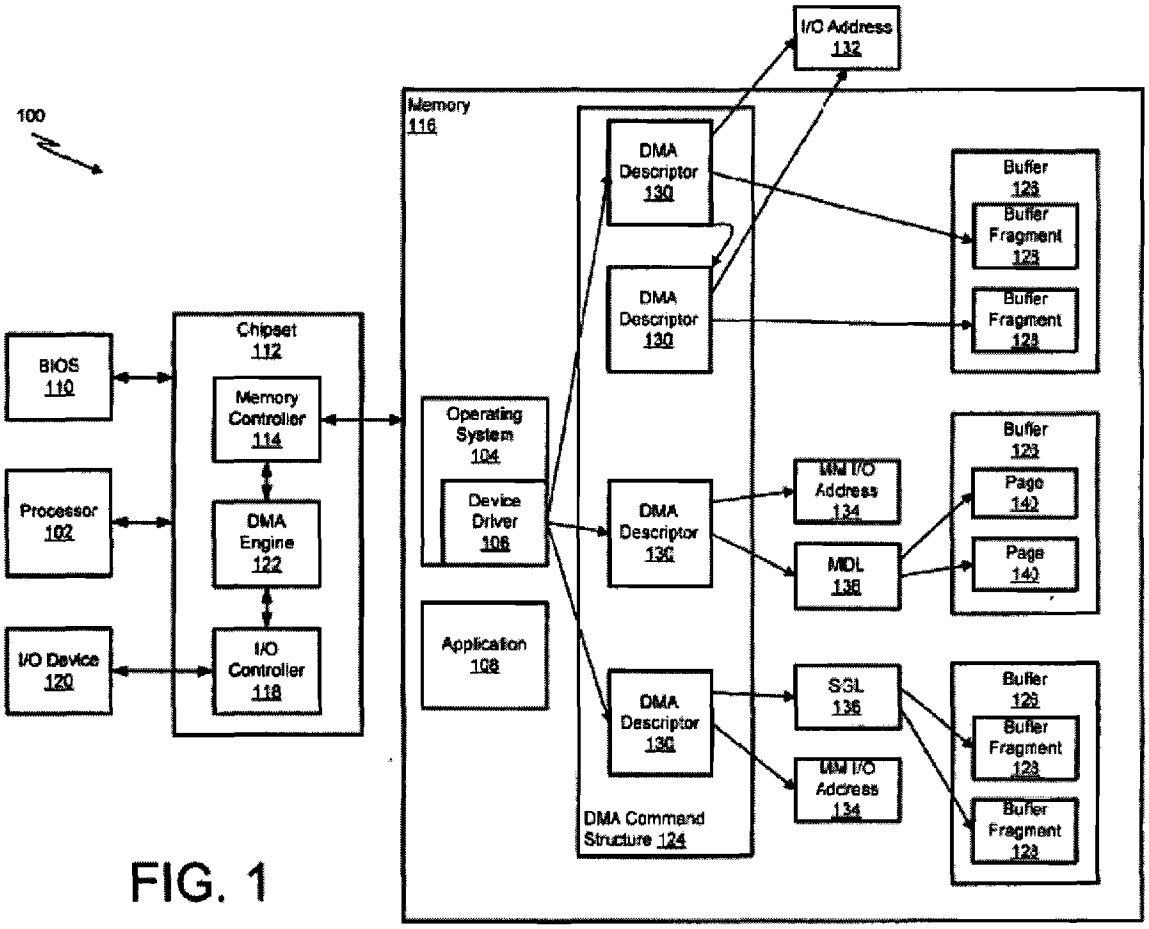


FIG. 1

As seen in the figure, reference number 116 represents a memory; however, there is no teaching in *Futral* of a **cache**.

The Final Office Action argues that the definition of "direct memory access (DMA)" is "memory access that does not involve the microprocessor and is frequently used for data transfer directly between memory and an 'intelligent' peripheral device," according to the Microsoft Computer Dictionary. The Final Office Action then concludes that the DMA controller of *Futral*

executes DMA commands for management of a cache, because the memory shown at 116 in FIG. 1 of *Futral* allegedly is a cache that is accessed by the DMA controller. Appellants respectfully disagree. The Final Office Action references Appellants' own definition of a cache as follows: "[a] cache is a storage that keeps frequently accessed data or program instructions readily available so that the device, in this case a DMA controller, does not access them repeatedly from slower storage." It would be clear to a person of ordinary skill in the art that memory 116 of *Futral* is a slower storage and not a cache. If memory 116 were a cache, then *Futral* would call it a "cache."

Furthermore, the applied reference cannot teach or fairly suggest the further limitation of a DMA controller that is configured to execute DMA commands for the management of a cache. *Futral* may teach a processor issuing DMA commands to a DMA controller to perform DMA operations; however, *Futral* simply does not teach or fairly suggest a processor issuing DMA commands to a DMA controller to perform **cache management** operations.

The applied reference fails to teach or suggest each and every claim limitation; therefore, *Futral* does not anticipate claims 1, 9, and 15. Since claims 3, 11, 16, and 17 depend from claims 1, 9, and 15, the same distinctions between *Futral* and the invention recited in claims 1, 9, and 15 apply for these claims. In addition, claims 3, 11, 16, and 17 recite further combinations of features not taught or suggested by the prior art.

Therefore, Appellants respectfully request that the rejection of claims 1, 3, 9, 11, and 15-17 under 35 U.S.C. § 102(e) not be sustained.

II. 35 U.S.C. § 103, Alleged Obviousness of Claims 2, 7, 8, 10, 13, 14, 20, and 21

The Office rejects claims 2, 7, 8, 10, 13, 14, 20, and 21 under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Futral* in view of *Ollivier et al.* (U.S. Patent No. 6,738,881). This rejection is respectfully traversed.

With respect to claims 2, 7, 8, 10, 13, 14, 20, and 21, the Final Office Action acknowledges that *Futral* does not teach a cache coupled to the DMA controller where the system is configured for the execution of the DMA commands for the management of a cache on the DMA controller to manage the operation of the cache coupled to the DMA controller. The Final Office Action alleges that *Ollivier* teaches a plurality of first-in-first-out (FIFO) buffers and

where the FIFOs are utilized to manage the data transferring to and from the plurality of memories.

Appellants submit that, like *Futral*, *Ollivier* fails to teach or suggest a cache or a DMA controller configured to execute DMA commands for cache management. While FIFO buffers may be used to transfer data, these buffers require no management. A buffer is not a cache. A cache is a storage that keeps frequently accessed data or program instructions readily available so the device, in this case a DMA controller, does not access them repeatedly from slower storage. Because of the nature of a cache, a device requires cache management to maintain the integrity of the data (or instructions) in the cache. On the other hand, a buffer is merely a one-time staging area for data being transferred. By its very nature, a FIFO buffer requires no management. That is, the first data in is the first data out. There is no cache coherency or data integrity problem with a FIFO buffer. Therefore, the system of *Ollivier* has no need for special DMA commands for cache management.

Neither *Futral* nor *Ollivier* teaches or suggests a cache or a DMA controller that is configured to execute DMA commands for cache management. Even assuming, *arguendo*, that a person of ordinary skill in the art would have found it obvious to combine *Futral* and *Ollivier*, the combination would not result in the invention recited in claims 2, 7, 8, 10, 13, 14, 20, and 21. Therefore, *Futral* and *Ollivier* do not render claims 2, 7, 8, 10, 13, 14, 20, and 21 obvious. Appellants respectfully request that the rejection of claims 2, 7, 8, 10, 13, 14, 20, and 21 under 35 U.S.C. § 103(a) not be sustained.

III. 35 U.S.C. § 103, Alleged Obviousness of Claims 4, 5, 12, 18, and 19

The Office rejects claims 4, 5, 12, 18, and 19 under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Futral* in view of *Liao et al.* (U.S. Patent No. 6,681,296). This rejection is respectfully traversed.

With respect to claims 4, 5, 12, 18, and 19, the Final Office Action acknowledges that *Futral* does not teach or suggest a DMA command that is a flush command or a zero command. The Final Office Action alleges that *Liao* teaches DMA commands that include a "block flush" command or a "block set to zero" command at col. 3, lines 8-23, which states:

> One strategy that has been used in the past in connection with caches to improve application performance is to provide in the

> instruction set of the microprocessor a mechanism that enables software assisted cache management. Most modern microprocessors provide instructions in the instruction set which enable software to assist the cache management hardware to some degree in managing the cache. For example, the PowerPC architecture contains several user-accessible instructions in the instruction set for manipulating the data cache that can significantly improve overall application performance. These instructions are: "block touch" (dcbt); "block touch for store" (dcbtst); "block flush" (dcbf); "block store" (dcbst); and "block set to zero" (dcbz). see Zen and the Art of Cache Maintenance, Byte Magazine, March 1997.

The above cited portion explicitly teaches a mechanism in the instruction set **of the microprocessor** that enables software assisted cache management. However, the *Liao* reference fails to teach or suggest providing **DMA commands** for cache management or a DMA controller that is configured to execute DMA commands for cache management. *Futral* also fails to teach or suggest a DMA controller that is configured to execute DMA commands for cache management. Even assuming, *arguendo*, that a person of ordinary skill in the art would have found it obvious to combine *Futral* and *Liao*, the combination would not result in the invention recited in claims 4, 5, 12, 18, and 19. Therefore, *Futral* and *Liao* do not render claims 4, 5, 12, 18, and 19 obvious. Appellants respectfully request that the rejection of claims 4, 5, 12, 18, and 19 under 35 U.S.C. § 103(a) not be sustained.

IV. 35 U.S.C. § 103, Alleged Obviousness of Claim 6

The Office rejects claim 6 under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Futral* in view of *Ohba* (U.S. Patent No. 6,427,201). This rejection is respectfully traversed.

With respect to claim 6, the Final Office Action acknowledges that *Futral* does not teach or suggest parameters of a DMA command comprising a tag. The Final Office Action alleges that *Ohba* teaches a system and method comprising a DMA packet including a tag-command (DMA-tag) at FIG. 6 and col. 7, line 64, to col. 8, line 3, are as follows:

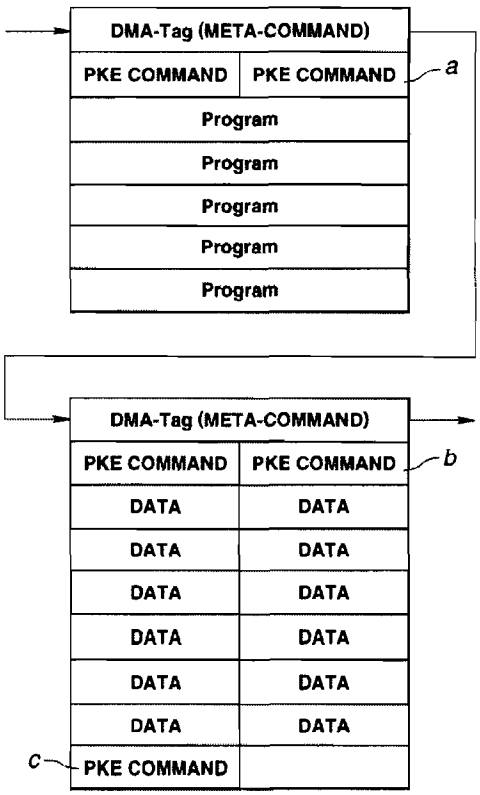


FIG.6

FIG. 6 shows an example of a DMA packet of the program and data.

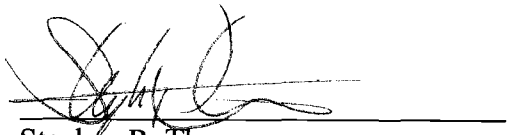
Referring to FIG. 6, there is arrayed, in this DMA command, a PKE command for the packet engine PKE as a data expansion mechanism, next to the tag-command (DMA-tag) to the DMAC 14, and is followed by a main portion of the program or data.

This cited portion does appear to teach a DMA-tag command. However, the *Ohba* reference fails to teach or suggest providing **DMA commands** for cache management or a DMA controller that is configured to execute DMA commands for cache management. *Futral* also fails to teach or suggest a DMA controller that is configured to execute DMA commands for cache management. Even assuming, *arguendo*, that a person of ordinary skill in the art would have found it obvious to combine *Futral* and *Ohba*, the combination would not result in the invention recited in claims 4, 5, 12, 18, and 19. Therefore, *Futral* and *Ohba* do not render claim 6 obvious. Appellants respectfully request that the rejection of claim 6 under 35 U.S.C. § 103(a) not be sustained.

CONCLUSION

In view of the above, Appellants respectfully submit that the features of claims 1-21 are not taught or suggested by the cited prior art references. Accordingly, Appellants request that the Board of Patent Appeals and Interferences overturn the rejections set forth in the Final Office Action.

Respectfully submitted,



Stephen R. Tkacs
Reg. No. 46,430
Walder Intellectual Property Law, P.C.
P.O. Box 832745
Richardson, TX 75083
(214) 722-6422
AGENT FOR APPELLANTS

CLAIMS APPENDIX

1. A system to provide software program control of cache management, comprising:

a processor configured to generate DMA commands for the management of a cache on the execution of a software program on the processor; and

a DMA controller coupled to the processor, configured to execute the DMA commands for the management of a cache.

2. The system of Claim 1, further comprising a cache coupled to the DMA controller, the system configured for the execution of the DMA commands for the management of a cache on the DMA controller to manage the operation of the cache coupled to the DMA controller.

3. The system of Claim 1, wherein at least one of the DMA commands is a get command and at least one of the DMA commands is a put command.

4. The system of Claim 1, wherein at least one of the DMA commands is a flush command.

5. The system of Claim 1, wherein at least one of the DMA commands is a zero command.

6. The system of Claim 1, wherein the parameters of the DMA commands comprise tag, transfer size, and effective address low.

7. The system of Claim 1, wherein the cache is a DMA cache tightly coupled to the DMA controller.

8. The system of Claim 1, wherein the cache is a cache for system memory.

9. A method for cache management in a system comprising a DMA controller and a processor, the method comprising the steps of:

running software on the processor to generate DMA commands for management of a cache;

issuing the DMA commands to the DMA controller; and

executing the DMA commands.

10. The method of Claim 9, wherein a cache is coupled to the DMA controller, and executing the DMA commands on the DMA controller manages the operation of the cache.

11. The method of Claim 9, wherein at least one of the DMA commands is a put command and wherein at least one of the DMA commands is a get command.

12. The method of Claim 9, wherein at least one of the DMA commands is a flush command.

13. The method of Claim 9, wherein the cache is a DMA cache tightly coupled to the DMA controller.

14. The method of Claim 9, wherein the cache is a cache for system memory.

15. A computer program product for cache management in a system comprising a DMA controller and a processor, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer code for running software on the processor to generate DMA commands for management of a cache;

computer code for issuing the DMA commands to the DMA controller; and

computer code for executing the DMA commands.

16. The computer program product of Claim 15, wherein at least one of the DMA commands is a get command.

17. The computer program product of Claim 15, wherein at least one of the DMA commands is a put command.

18. The computer program product of Claim 15, wherein at least one of the DMA commands is a flush command.

19. The computer program product of Claim 15, wherein at least one of the DMA commands is a zero command.

20. The computer program product of Claim 15, wherein the cache is a DMA cache tightly coupled to the DMA controller.

21. The computer program product of Claim 15, wherein the cache is a cache for system memory.

EVIDENCE APPENDIX

NONE

RELATED PROCEEDINGS APPENDIX

NONE